# Red Hat Process Automation Manager 7.8

## Integrating Red Hat Fuse with Red Hat Process Automation Manager

Last Updated: 2021-02-22

# Red Hat Process Automation Manager 7.8 Integrating Red Hat Fuse with Red Hat Process Automation Manager

Red Hat Customer Content Services
brms-docs@redhat.com

## Legal Notice

## Abstract

This document describes how to integrate Red Hat Process Automation Manager 7.8 with Red Hat Fuse on Apache Karaf and Red Hat JBoss Enterprise Application Platform.

# Table of Contents

# PREFACE

As a system administrator, you can integrate Red Hat Process Automation Manager with Red Hat Fuse on Apache Karaf and Red Hat JBoss Enterprise Application Platform to facilitate communication between integrated services.

NOTE

Red Hat Process Automation Manager does not support integration with Red Hat Fuse on Spring Boot.

# CHAPTER 1. RED HAT FUSE AND RED HAT PROCESS AUTOMATION MANAGER

Red Hat Fuse is a distributed, cloud-native integration platform that is part of an agile integration solution. Its distributed approach enables teams to deploy integrated services where required. Fuse has the flexibility to service diverse users, including integration experts, application developers, and business users, each with their own choice of deployment, architecture, and tooling. The API-centric, container-based architecture decouples services so they can be created, extended, and deployed independently. The result is an integration solution that supports collaboration across the enterprise.

Red Hat Process Automation Manager is the Red Hat platform for creating business automation applications and microservices. It enables enterprise business and IT users to document, simulate, manage, automate, and monitor business processes and policies. Red Hat Process Automation Manager is designed to empower business and IT users to collaborate more effectively, so business applications can be changed easily and quickly.

You can install Red Hat Fuse on the Apache Karaf container platform or Red Hat JBoss Enterprise Application Platform and then install and configure Red Hat Process Automation Manager in that container.

IMPORTANT

For the version of Red Hat Fuse that Red Hat Process Automation Manager 7.8 supports, see Red Hat Process Automation Manager 7 Supported Configurations .

## 1.1. RED HAT PROCESS AUTOMATION MANAGER KARAF FEATURES

The following table lists Red Hat Process Automation Manager Karaf features.

| Feature | Description |
|---|---|
| **drools-module** | Contains the core and compiler of Drools, used to create KIE bases and KIE sessions from plain DRL. It also contains the implementation of the executable model. Uses Drools for rules evaluation, without requiring persistence, processes, or decision tables. |
| **drools-template** | Contains the Drools templates. |
| **drools-jpa** | Uses Drools for rules evaluation with persistence and transactions, but without requiring processes or decision tables. The **drools-jpa** feature includes the **drools-module**, however you might also need to install the **droolsjbpm-hibernate** feature, or ensure that a compatible hibernate bundle is installed. |
| **drools-decisiontable** | Uses Drools with decision tables. |

| Feature | Description |
|---|---|
| **jbpm** | Uses jBPM. The **jbpm** feature includes the**drools-module** and **drools-jpa**. You might need to install the **droolsjbpm-hibernate** feature, or ensure that a compatible hibernate bundle is installed. |
| **jbpm** and **jbpm-human-task** | Uses jBPM with human tasks. |
| **jbpm-workitems-camel** | Provides the **jbpm-workitems-camel** component. |
| Core engine JARs and **kie-ci** | Uses Red Hat Process Automation Manager with the KIE scanner (**kie-ci**) to download kJARs from a Maven repository. |
| **kie-camel** | Provides the **kie-camel** component, an Apache Camel endpoint that integrates Fuse with Red Hat Process Automation Manager. |
| **kie-spring** | Installs the **kie-spring** component that enables you to configure listeners to KIE sessions using XML tags. |

# CHAPTER 2. RED HAT PROCESS AUTOMATION MANAGER DECISION AND PROCESS ENGINES WITH FUSE ON APACHE KARAF

Apache Karaf is a standalone open-source runtime environment. It is based on the OSGi standard from the OSGi Alliance. Karaf provides support for modularisation through OSGi bundles with sophisticated class-loading support. You can deploy multiple versions of a dependency side by side in a Karaf container. You can use hot code swapping to upgrade or replace a module without shutting down the container.

Red Hat Process Automation Manager integration with Fuse on Karaf is provided through Karaf features. You can install individual components of Red Hat Process Automation Manager for Fuse on Karaf using these features.

Features files are XML files that specify which OSGI bundles are installed for a particular feature. The following features XML files facilitate Red Hat Process Automation Manager and Fuse on Karaf integration:

- **rhba-features-<FUSE-VERSION>-features.xml**
  This file is part of Fuse installed in Karaf where **<FUSE-VERSION>** is the version of Fuse. This file is stored in the Karaf system repository, in the **system/org/jboss/fuse/features/rhba-features** directory. This file contains prerequisites for installing Red Hat Process Automation Manager features.

- **kie-karaf-features-7.39.0.Final-redhat-00005-features-fuse.xml**
  This file is part of Red Hat Process Automation Manager and provides Red Hat Process Automation Manager features, which define the OSGi features that can be deployed into Red Hat Fuse. OSGi users can install features from this file to install Red Hat Process Automation Manager into Fuse and use it in their applications. You can find this features file in the online and offline Maven repository that is distributed with Red Hat Process Automation Manager. The group ID, artifact ID, and version (GAV) identifier of this file is **org.kie:kie-karaf-features:7.39.0.Final-redhat-00005**.

## 2.1. UNINSTALLING OBSOLETE RED HAT PROCESS AUTOMATION MANAGER FEATURES XML FILES ON KARAF

If your installation contains older versions of the Red Hat Process Automation Manager features XML files (for example, **kie-karaf-features-<VERSION>-features.xml**), you must remove these files and all associated files before installing the most recent features XML files.

### Prerequisites

- Obsolete features XML files exist in your Apache Karaf installation.

### Procedure

1. Enter the following commands to determine whether your installation contains obsolete Red Hat Process Automation Manager features XML files:

```
$ JBossFuse:karaf@root> feature:repo-list
$ JBossFuse:karaf@root> feature:list
```

2. Enter the following command, where **<FUSE_HOME>** is the Fuse installation directory, to start the Red Hat Fuse console:

   ```
   $ ./<FUSE_HOME>/bin/fuse
   ```

3. Enter the following command, where **<FEATURE_NAME>** is the name of the feature that you want to uninstall, to uninstall features or applications that use obsolete features XML files:

   ```
   JBossFuse:karaf@root> features:uninstall <FEATURE_NAME>
   ```

   The following example shows how to remove features:

   ```
   JBossFuse:karaf@root> features:uninstall drools-module
   JBossFuse:karaf@root> features:uninstall jbpm
   JBossFuse:karaf@root> features:uninstall kie-ci
   ```

4. Search Karaf home for references to bundles that use **drools**, **kie**, or **jbpm**. The following example shows how to use **grep** to search for these components:

   ```
   karaf@root> list -t 0 -s | grep drools
   karaf@root> list -t 0 -s | grep kie
   karaf@root> list -t 0 -s | grep jbpm
   ```

   The example shows output from these commands:

   ```
   250 │ Active   │ 80 │ 7.19.0.201902201522          │ org.drools.canonical-model
   251 │ Active   │ 80 │ 7.19.0.201902201522          │ org.drools.cdi
   252 │ Active   │ 80 │ 7.19.0.201902201522          │ org.drools.compiler
   ```

5. Enter the following command, where **BUNDLE_ID** is a bundle ID returned in the search, to remove the bundles found in the previous step:

   ```
   karaf@root> osgi:uninstall BUNDLE_ID
   ```

6. Enter the following command to remove the obsolete **drools-karaf-features** URL:

   ```
   karaf@root> features:removeurl
    mvn:org.kie/kie-karaf-features/VERSION.Final-redhat-VERSION/xml/features
   ```

7. Restart Fuse.

## 2.2. INSTALLING RED HAT PROCESS AUTOMATION MANAGER FEATURES ON KARAF USING XML FILES

Install Red Hat Process Automation Manager features on Karaf to create a dynamic runtime environment for your Red Hat Process Automation Manager processes.

### Prerequisites

- A Red Hat Fuse installation in an Apache Karaf container is available. For information about installing Fuse in Apache Karaf, see *Installing Red Hat Fuse on the Apache Karaf container* .

- You have removed any obsolete Red Hat Process Automation Manager features XML files as described in Section 2.1, “Uninstalling obsolete Red Hat Process Automation Manager features XML files on Karaf”.

### Procedure

To install Red Hat Process Automation Manager features, enter the following command:

```
$ JBossFuse:karaf@root> feature:install <FEATURE_NAME>
```

#### NOTE

Use **org.drools.osgi.spring.OsgiKModuleBeanFactoryPostProcessor** instead of **org.kie.spring.KModuleBeanFactoryPostProcessor** to postprocess KIE elements in an OSGi environment.

Do not install the **drools-module** feature before the **kie-spring** feature. If you do, the **drools-compiler** bundle will not detect packages exported by **kie-spring**.

If you install the features in the incorrect order, run **osgi:refresh drools-compiler_bundle_ID** to force the **drools-compiler** to rebuild its **Import-Package** metadata.

In this command, **<FEATURE_NAME>** is one of the features listed in Section 1.1, “Red Hat Process Automation Manager Karaf features”.

## 2.3. INSTALLING RED HAT PROCESS AUTOMATION MANAGER FEATURES ON KARAF THROUGH MAVEN

Install Red Hat Process Automation Manager with Fuse on Apache Karaf to deploy integrated services where required.

### Prerequisites

- A Red Hat Fuse 7.6 on Apache Karaf installation exists. For installation instructions, see *Installing Red Hat Fuse on the Apache Karaf container* .
- Any obsolete features XML files have been removed, as described in Section 2.1, “Uninstalling obsolete Red Hat Process Automation Manager features XML files on Karaf”.

### Procedure

1. To configure the Maven repository, open the **FUSE_HOME/etc/org.ops4j.pax.url.mvn.cfg** file in a text editor.

2. Make sure that the **https://maven.repository.redhat.com/ga/** repository is present in the **org.ops4j.pax.url.mvn.repositories** variable and add it if necessary.

   #### NOTE

   Entries are separated by a comma, space, and backslash (, \). The backslash forces a new line.

3. To start Fuse, enter the following command, where **FUSE_HOME** is the Fuse installation directory:

   ```
   $ ./FUSE_HOME/bin/fuse
   ```

4. To add a reference to the features file that contains installation prerequisites, enter the following command, where **<FUSE_VERSION** is the version of Fuse that you are installing:

   ```
   $ feature:repo-add mvn:org.jboss.fuse.features/rhba-features/<FUSE-
   VERSION>/xml/features
   ```

5. Enter the following command to add a reference to the Red Hat Process Automation Manager features XML file:

   ```
   $ JBossFuse:karaf@root> features:addurl mvn:org.kie/kie-karaf-
   features/VERSION/xml/features-fuse
   ```

   To see the current **drools-karaf-features** version, see the Red Hat Process Automation Manager 7 Supported Configurations page.

6. Enter the following command to install a feature provided by Red Hat Process Automation Manager features XML file. In this command, **<FEATURE_NAME>** is one of the features listed in Section 1.1, “Red Hat Process Automation Manager Karaf features” .

   ```
   JBossFuse:karaf@root> features:install <FEATURE_NAME>
   ```

7. Enter the following command to verify the installation:

   ```
   $ JBossFuse:karaf@root>feature:list
   ```

   Successfully installed features have the status **started**.

# CHAPTER 3. INSTALLING FUSE ON RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

Install Red Hat Fuse 7.6 on Red Hat JBoss EAP 7.2.3 to use with Red Hat Process Automation Manager to deploy integrated services where required.

> **NOTE**
>
> Red Hat Fuse 7.6 is supported on Red Hat JBoss EAP 7.2.3, whereas Red Hat Process Automation Manager is supported on Red Hat JBoss EAP 7.3. To ensure a stable deployment environment, install Fuse and Red Hat Process Automation Manager on different instances of Red Hat JBoss EAP according to the supported Red Hat JBoss EAP versions.

### Prerequisites

- A Red Hat Process Automation Manager installation on Red Hat JBoss Enterprise Application Platform 7.3 is available. For installation instructions, see *Installing and configuring Red Hat Process Automation Manager on Red Hat JBoss EAP 7.3*.
- A separate instance of Red Hat JBoss Enterprise Application Platform 7.2.3 is available.

### Procedure

1. Install Red Hat Fuse 7.6 on Red Hat JBoss Enterprise Application Platform 7.2.3. For installation instructions, see the *Install Fuse 7.6 on JBoss EAP 7.2* .
2. Open the **pom.xml** file in the Fuse home directory in a text editor.
3. Create the integration project with a dependency on the **kie-camel** component by editing the **pom.xml** file as shown in the following example:

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-core</artifactId>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
  <exclusions>
    <exclusion>
      <groupId>aopalliance</groupId>
      <artifactId>aopalliance</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.kie.server</groupId>
  <artifactId>kie-server-api</artifactId>
```

```
   <exclusions>
     <exclusion>
       <groupId>org.jboss.spec.javax.xml.bind</groupId>
       <artifactId>jboss-jaxb-api_2.3_spec</artifactId>
     </exclusion>
     <exclusion>
       <groupId>javax.activation</groupId>
       <artifactId>activation</artifactId>
     </exclusion>
   </exclusions>
</dependency>
<dependency>
   <groupId>org.jbpm</groupId>
   <artifactId>jbpm-bpmn2</artifactId>
 </dependency>
 <dependency>
   <groupId>org.kie</groupId>
   <artifactId>kie-camel</artifactId>
   <exclusions>
     <exclusion>
       <groupId>org.apache.cxf</groupId>
       <artifactId>cxf-core</artifactId>
     </exclusion>
     <exclusion>
       <groupId>org.apache.camel</groupId>
       <artifactId>camel-cxf</artifactId>
     </exclusion>
     <exclusion>
       <groupId>org.apache.camel</groupId>
       <artifactId>camel-cxf-transport</artifactId>
     </exclusion>
     <exclusion>
       <groupId>com.thoughtworks.xstream</groupId>
       <artifactId>xstream</artifactId>
     </exclusion>
   </exclusions>
</dependency>
<dependency>
    <groupId>org.kie.server</groupId>
    <artifactId>kie-server-client</artifactId>
    <exclusions>
      <exclusion>
        <groupId>org.jboss.spec.javax.ws.rs</groupId>
        <artifactId>jboss-jaxrs-api_2.0_spec</artifactId>
      </exclusion>
    </exclusions>
</dependency>
```

# CHAPTER 4. THE KIE-CAMEL COMPONENT

The **kie-camel** component is an Apache Camel endpoint provided by Red Hat Fuse that integrates Fuse with Red Hat Process Automation Manager. It enables you to specify a Red Hat Process Automation Manager module by using a Maven group ID, artifact ID, and version (GAV) identifier which you can pull into the route and execute. It also enables you to specify portions of the message body as facts.You can use the **kie-camel** component with embedded engines or with KIE Server.

## Embedded engines

In this scenario, KIE engines run in the same container as the Fuse integration project and you can communicate with engines using KIE commands. To create the Camel producer, use the following URI:

```
kie-local:kie-session-name?action=execute
```

For example, enter the following command to initialize a Camel route in Spring:

```
<from uri="direct:runCommand" />
  <to uri="kie-local:kie-session1?action=execute"/>
```

## KIE Server

In this scenario, the **kie-camel** component connects to KIE Server using the KIE Server REST API. This enables users to communicate with the KIE Server using the KIE Server API. To create a producer, use the following URI:

```
kie:http://username:password@kie-server-url`
```

For example, enter the following command to initialize a Camel route in Spring:

```
<from uri="direct:runCommand" />
  <to uri="kie:http://user:psswd@localhost:8080/kie-server-services/services/rest/server"/>
```

The message has the following headers:

**Table 4.1. Message headers and descriptions**

| Header | Description |
|---|---|
| **CamelKieClient** | KIE Server client (mandatory) |
| **CamelKieOperation** | KIE Server client (mandatory) |
| **CamelKieParameterName** | The value of the client method parameter (optional) |
| **CamelKieBodyParam** | The method parameter where the message body is stored (optional) |

# APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Thursday, January 21, 2021.